Journal of AI and Data Mining
Vol 7, No 4, 2019, 607-616
DOI: 10.22044/JADM.2019.7183.1852

A Novel Architecture for Detecting Phishing Webpages using Cost-based Feature Selection

A. Zangooei[1], V. Derhami[1] and F. Jamshidi[2*]

1. Computer Engineering Department, Faculty of Engineering, Yazd University, Yazd, Iran.
2. Department of Electrical Engineering, Faculty of Engineering, Fasa University, Fasa, Iran.

Received 23 June 2018; Revised 21 January 2019; Accepted 07 March 2019.
**Corresponding author: jamshidi@fasau.ac.ir (F. Jamshidi).*

Abstract
Phishing is one of the luring techniques used to exploit personal information. A phishing webpage detection system (PWDS) extracts features to determine whether it is a phishing webpage or not. Selecting appropriate features improves the performance of PWDS. The performance criteria are detection accuracy and system response time. The major time consumed by PWDS arises from feature extraction, which is considered as feature cost in this paper. Here, two novel features are proposed. They use the semantic similarity measure to determine the relationship between the content and the URL of a page. Since the suggested features do not apply third-party services such as search engine result, the feature extraction time decreases dramatically. Login form pre-filer is utilized to reduce unnecessary calculations and false positive rate. In this paper, a cost-based feature selection is presented as the most effective feature. The selected features are employed in the suggested PWDS. The extreme learning machine algorithm is used to classify webpages. The experimental results demonstrate that the suggested PWDS achieves a high accuracy of 97.6% and a short average detection time of 120.07 ms.

Keywords: *Cost-based feature selection, Extreme learning machine, Phishing, Semantic similarity, Term Frequency and Inverse Document Frequency.*

1. Introduction

Generally, fraud "is the act of deceiving to gain unfair, undeserved and/or illegal financial profit" [1]. The prominent role of the internet in businesses provides a strong motivation for attackers to commit frauds. One of these frauds is the phishing attack. Phishing is an attempt to obtain sensitive information such as the username, password, and credit card details through a foregoing webpage or an E-mail address. A Phishing website is mock and looks similar to the page of a real website. Most phishing attacks start with an electronic letter claiming issued by a reputable company [2]. This E-mail encourages the user to click on the address provided in its content. This address directs the user to an illegal webpage that is designed similar to a valid website. The targets of phishing are the popular and online payment websites. Recently, phishing threats have increased rapidly [3]. The Anti-Phishing Working Group has reported 44,407 unique phishing websites in 2014 [4]. The financial loss imposed on worldwide organizations in 2014 has been estimated at $453 million [5]. These alarming trends have resulted in the loss of consumers' trust in using E-commerce websites [6].The existing PWDSs have a high response time, and, in some cases, extract undesirable features, so the phishing webpages are not detected. PWDS identifies the phishing webpages based on the features extracted from them in the shortest possible time. Therefore, features with a high detection efficiency and a short response time should be extracted.

In this paper, two novel features are suggested that determine the relationship between the content and the URL of a page. One of them uses a semantic similarity measure to obtain the similarities

between page textual content and page address. According to the relationship between the link and address of the page, the similarities between the link's text and the page's address is calculated as the other feature. The proposed features are independent from third-party services. Thus the feature extraction time is short. A cost-based feature selection is suggested to choose more effective features of the phishing problem with a short extraction time. The Minimal-Redundancy-Maximal-Relevance (MRMR) feature selection method is converted to cost-based MRMR feature selection. Extreme Learning Machine (ELM) algorithm classifies and detects phishing pages.

The rest of this paper is organized as what follows. Section 2 presents an overview of the related works. Section 3 illustrates the overall system architecture. Sections 4, 5, and 6 explain the methodology used in keyword, URL identity extractors, and webpage feature generator, respectively. Section 7 describes the proposed cost-based MRMR feature selection. Section 8 presents the features applied in the suggested PWDS. Sections 9 and 10 explain page classifier and login form finder pre-filter, respectively. The experimental results are discussed in Section 11. Conclusion and future works are presented in Section 12.

2. Related works

White-list approaches have a list of legitimate websites and their associated information including IP address. This list needs to be updated constantly. Websites not existing in this list are suspicious. In [7], Phishing Guard is proposed to prevent access to phishing websites with a URL similarity check. Access enforcement facility checks if the site is safe. The black-list approaches have a list of phishing sites. Websites not existing in this list are safe. In [8], PhishNet has been proposed. It uses an approximate matching algorithm to divide a URL into multiple components, and compares them with entries in the black-list.

The visual similarity-based approaches detect phishing attacks by comparing visual and image similarities between the webpages. In [9], an approach has been proposed to convert the webpages into normalized images, and use color and coordinate features to represent the image signature. The earth mover distance calculates the signature distance of the image and trained threshold vector for classifying a webpage as phishing or legitimate. In [10], a method has been proposed to decompose the webpage into block regions depending on ‘‘visual cues.’’ The visual similarity between two webpages is measured using the block level, layout, and style similarities. A webpage is phishing if any metric value is higher than a threshold. In [11], a system has been proposed to identify phishing webpages, considering text pieces and their style, images embedded, and the overall visual appearance of the page the user sees.

Heuristic-based approaches extract features from a webpage to detect phishing. The features are extracted from the page address, content, and HTML DOM of the webpage. In [12], a TF-IDF (term frequency and inverse document frequency) based algorithm has been proposed to identify keywords from the webpage. These keywords are searched on Google. A webpage is legitimate if the page domain exists in the top N search results. In [3], a three-layered architecture is developed for an anti-phishing system using the web service technologies. This model has three layers. The first layer is a client interface that extracts the URL requested by users from a web browser and sends it to phishing verification. The middle layer communicates between the client interface layer and the third layer that uses a set of heuristics and resources on the internet to detect phishing webpage. In [13], a hybrid anti-phishing approach that has three modules is presented. The first two modules are pre-filters to reduce false positives. In the third module, 15 pivotal heuristics and SVM classifier are used for checking the phishiness of webpages. In [14], a method has been proposed to detect phishing webpages based on its content, HTTP transaction, and search engine results. This method uses SVM for classification. If a website claims a fake identity, then it is classified as phishing.

In [15], a layered anti-phishing approach has been proposed. It uses URL features utilizing the HTML DOM, search engine, and third-party services, and finally, machine learning techniques for classification. It applies two pre-filters to decrease false positives. In [2], two feature sets have been proposed. One uses Levenshtein Distance for string matching to find the relationship between the content and the URL of a webpage, and the other identifies the access protocol of page resource elements.

3. System architecture

Figure 1 shows our proposed architecture system. It consists of three phases: (1) pre-processing phase, (2) training phase, and (3) testing phase. In this paper, the training and testing phases are known as PWDS. The pre-processing phase is applied to select the best relevant features that improve the accuracy and response time of PWDS.

The most effective features are chosen using the suggested cost-based feature selection. The process is as follows: the keyword and URL identity sets are extracted from the webpage. The identity extraction methods are described in Sections 4 and 5. Fifty-two features are extracted from the webpages. Feature values and their extraction times are stored for each webpage as a dataset. The proposed cost-based feature selection chooses the best relevant features. These features are used in the suggested PWDS. In the training phase, the feature values are obtained for each instance of the training corpus. The machine learning engine utilizes them to build the classifier. In the test phase, the classifier uses a label to show whether a real webpage is a phish or not. Here, the login form detector pre-filter is applied to check whether a login field exists in a given web page or not. If a login form is found in the webpage, the URL identity set and the relevant features will be extracted from the webpage. The feature values are gathered to generate a feature vector. This vector is passed onto the phishing classifier for class identification.

4. Keyword identity (KI) extractor

The KI set of a webpage is a set of important words in it. The KI set is usually extracted by considering Body tag, Title tag, Meta description tag, Meta keyword tag, Alt attribute of all tags, and Title attribute of all tags of a webpage [12]. The KI set extraction method is taken from [12]. In this paper, the TF-IDF method indicated in [3] is applied to extract the KI set.

5. URL identity (URLI) extractor

The URLI set of a webpage is determined by analyzing its hyperlink structure. In legitimate websites, most of the links point to its own or associated domain but in the phishing pages, most of the links point to a foreign domain in order to imitate the behavior of a legitimate page [3]. In this work, the method indicated in [3] is applied to extract the URLI set.

6. Webpage feature generator

In the page address and HTML source code, there are many features that can distinguish the original legitimate website from the forged websites. The URLI and KI sets are used by the features in this module. Here, 52 features are extracted from the webpage. The features 1-9, 10-15, 16-20, 21-31, 32-37, and 38-50 are taken from [13], [15], [14], [2], [16], and [3], respectively. The features 51 and 52 are the novel ones proposed in this paper.

7. Proposed cost-based MRMR feature selection

A user is sometimes interested in both improving a subset of features and reducing the associated costs. Here, the cost function of PWDS is the time consumed in extracting the features from the webpage. Improving the performance of classification is important as well. To achieve these goals, a feature subset with an acceptable classification performance and a shorter extraction time is applied. MRMR is a ranker filter algorithm consisting of minimal redundancy and maximal relevance. The mutual information between the two discrete variables x and y that measure the similarity between features and correlation between feature and class is calculated as:

$$I(x,y)=\sum_{i,j}p(x_i,y_i)\log\frac{p(x_i,y_i)}{p(x_i)p(y_i)} \qquad (1)$$

where p stands for probability.

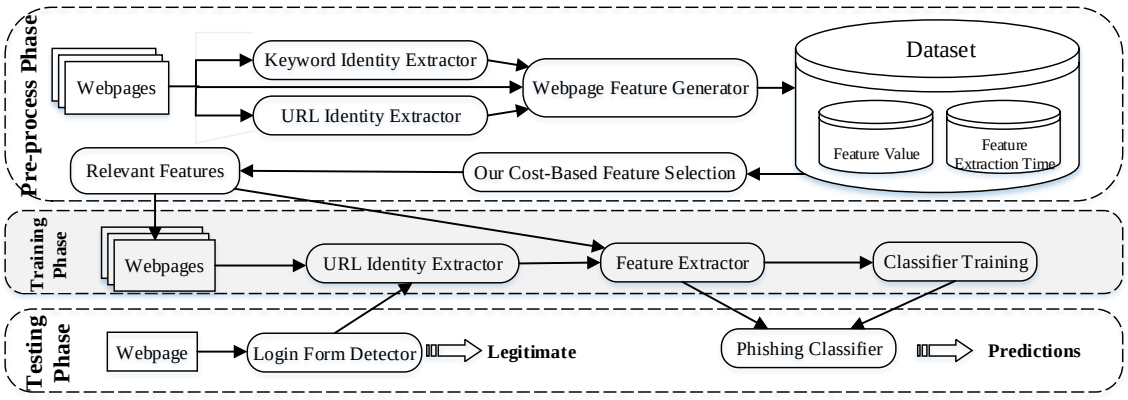

Figure 1. Proposed architecture system.

The idea of minimum redundancy is to select the features that are mutually maximally dissimilar [17]. W denotes the minimum redundancy, and is as follows:

$$W = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \quad (2)$$

where S is a set of features and $I(x_i, x_j)$ is the mutual information between features x_i and x_j. V denotes the maximal relevance, and is the mean value of all mutual information values between individual x_i and class c. It is defined as follows:

$$V = \frac{1}{|S|} \sum_{x_i \in S} I(x_i, c) \quad (3)$$

where $I(x_i, c)$ is the mutual information between x_i and c. In this method, the first feature has the highest $I(x_i, c)$. The rest of the features are chosen in an incremental way.

If S_{m-1} stands for the feature set with $m-1$ features, the mth feature will be selected from the set $X - S_{m-1}$ that maximize (4) for every $x_j \in X - S_{m-1}$:

$$I(x_j, c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j, x_i) \quad (4)$$

To convert MRMR to cost-based MRMR, a condition is added to the algorithm that examines the cost of the selected features. The new algorithm is called CBMRMR. Assume that S_{m-1} exists; when the evaluation function determines the m th feature, the MRMR algorithm continues as follows: the m th selected feature is added to S temporarily. The sum of the feature costs of S is calculated. The following condition is checked:

$$\sum_{x_i \in S} cost(x_i) < \gamma \quad (5)$$

where $cost(x_i)$ represents the cost of feature x_i, and γ is the threshold value defined by the user. Actually, the sum of the costs of features selected by the algorithm should be less than γ. If (5) is satisfied, the feature is added permanently to S. Otherwise, feature m is removed from S. As stated earlier, the first feature selected by MRMR is the feature with the greatest $I(x_i, c)$ value. After selecting the first feature, the condition in (5) is examined, and if the cost of the feature exceeds γ, the feature is ignored and another feature with the greatest $I(x_i, c)$ value is selected. The procedure continues until the first feature satisfies (5), and the rest of the algorithm continues with the aforementioned procedure.

8. Relevant features

The feature set chosen by the CBMRMR feature selection includes 17 features, which are described as follow. The first and second features are the proposed innovations of this paper.

Feature 1. Relation between content and URL of the webpage: In most legitimate webpages, the contents and the page URL are related. The address "https://login.yahoo.com" is related. "Yahoo" or the address "https://personal.co-operativebank. co.uk" is related to the "co-operative" bank. In legitimate webpage, titles and metadescription and keyword tags clearly indicate its content. "iCloud-Apple" is the title for the page "https://www.apple.com/icloud/", which indicates that iCloud page is associated with Apple. In contrast, the phishing pages do not use this feature. The page http://mezzafoods. net.au/usaa.com-sec-inetauth-logon/ entitled "USSA/welcome to USSA" is the phishing page of the USSA website. For page P, the relationship is computed as follows: Title, meta description, and keyword tags are extracted from P. Texts are converted into a list of words with lower case letters. Words with lengths equal or smaller than two characters and stop words are removed. Stemmed words and repetitive words are removed as well. In some pages, abbreviations of words are used. Examples are Kelley Blue Book (kbb.com). Thus the string that results from first letters is added to the list. List T with length l is generated.

The second-level domain, D_p, and the sub-domain, SD_p, are extracted from the URL of P. Some domains may be composed of several words attached to each other. For example, D_p of "https://www.timeanddate.com" must be decomposed to "time, and, date". To decompose it, Word segmentation 0.3.5 is used to turn D_p and SD_p into small lists named DW and SDW with lengths n and m, respectively. The direct relationship between the content and D_p / SD_p are defined, respectively, as (6) and (7):

$$A_1 = CMatch(T_i, D_p) \quad (6)$$

$$A_2 = CMatch(T_i, SD_p) \quad (7)$$

where T_i is the ith word in list T. $CMatch$ is a function that checks direct inclusion or absence of T_i in D_p and return values of either one or zero, respectively.

Semantic similarity between the content and D_p / SD_p are computed, respectively, as (8) and (9):

$$A_3 = \max_{i \in l,\, j \in n} \left(wup\left(T_i, DW_j\right)\right) \quad (8)$$

$$A_4 = \max_{i \in l,\, j \in m} \left(wup\left(T_i, SDW_j\right)\right) \quad (9)$$

where $wup(a,b)$ computes the Wu and Palmer semantic similarity [18] for strings a and b . DW_j is the j th word in list DW , and SDW_j is the j th word in list SDW .

Combining the direct and semantic relationships obtained, the $F1$ value is as:

$$F1 = \sum_{i=1}^{4} A_i \quad (10)$$

Feature 2. Relation between anchor text and URL of the webpage: In legitimate webpages, URL and text of anchor are related to the page URL. Most of the links in the electronic banking websites are related to this issue. Thus a feature that calculates the relation between the anchor text and the page URL is suggested in this paper. For page P , all anchor texts (the text between <a> tag) are extracted. The texts are converted to a list of terms with lower case letters. The terms with lengths equal or smaller than two characters and stop words are removed. Stemmed words and repetitive terms are removed, and list S is generated. TF-IDF is calculated on each term of S and the terms are ranked in list K . The top ten terms are retrieved from K as list T . After that, like the first feature, D_p , SD_p , DW , and SDW are generated. Finally, the $F2$ value is calculated using (6)-(10).

Feature 3. Login form with invalid action field: The authentication methods of legitimate websites are usually called via URLs in the action field of the HTML form [15]. In the phishing sites, the action field of the HTML form is typically empty or a simple file name. $F3 = 1$ if the action field of login form is empty or a simple file; otherwise, $F3 = 0$.

Feature 4. SSL certificate: Using the security protocols is one of the essential requirements for internet banking sites [19]. An attacker may apparently use this by changing the browser's address or using non-credit certificates or self-signed certificates [2]. $F4 = 1$ if the webpage address uses SSL certificate; otherwise, $F4 = 0$.

Feature 5. Foreign request identity: Phishing pages might request images, Javascript, and other objects from real websites [14]. A foreign request is a request that points to a foreign domain. To verify the identity of the foreign requests, first, the domain is extracted for each foreign request (e.g. File, Image, and Script). Secondly, the domain is compared with URLI and KI. Any matches would increase R_{id} by one. If $R_f > 0$, then $F5 = R_{id} / R_f$, where R_f is the number of all requests; otherwise, $F5 = 0$.

Feature 6. Dots in page address: The address of a phishing page may have many dots to confuse the users. If URL has more than five dots, $F6 = 1$; otherwise, $F6 = 0$.

Feature 7. Login-form identity: The most common technique used in a phishing attack is to bait users to reveal their credentials through login forms on a fake website [13]. This feature checks the legitimacy of login forms. In legitimate webpages, there is a relationship between value of the action attribute and KI and URLI. Domain name of the action value (D_{action}) is extracted, and then D_{action} is compared with KI and URLI. If D_{action} is not a foreign domain and URLI is a foreign domain and any of keywords in KI is a part of D_{action} , then the form result is 0; otherwise, 1. If D_{action} is a foreign domain, then the result is 1. If D_{action} is a foreign domain and URLI is a local domain, then the result is -1.

If the action attribute does not exist or the value is null, then the result is 0. The algorithm will be repeated for every login form on a webpage. If any form result is 1 or 0, then $F7$ is the same (1 or 0); otherwise, $F7 = -1$.

Feature 8. Foreign anchors: A foreign anchor points to a foreign domain. A webpage with a majority of foreign anchors is suspicious [3]. If $A_a > 0$, $F8 = A_f / A_a$, A_f is the number of foreign anchors and A_a is the number of all anchors; otherwise, $F8 = 0$.

Feature 9. Domain name in the path part of the page address: Some phishing URLs add the domain name of a legitimate website within the path segment of URLs to trick users [13]. Top-level domains, second-level domains, and all combinations are stored in a list. $F9 = 1$ if any of the entries in the list matches a part of the path segment; otherwise, $F9 = 0$. If there is no path segment in URL, $F9 = -1$.

Feature 10. IP address: Many phishing webpages use IP addresses instead of domain or host names. $F10=1$ if the webpage address contains IP address; otherwise, $F10=0$.

Feature 11. The '@' symbol in a domain name: This feature checks the existence of the "@" symbol in the domain name. The string to the left of '@' is treated as the "user info", and the string to the right is treated as the actual domain for retrieving a page. $F11=1$ if the domain name includes'@' character; otherwise, $F11=0$.

Feature 12. Nil anchors: A nil anchor points to nowhere, and the value of the href attribute of <a> tag will be null. A page with more nil anchors is more suspicious [3]. If $A_a>0$, then $F12=A_{nil}/A_a$, A_{nil}, where is the number of nil anchors; otherwise, $F12=0$.

Feature 13. Phishing Keywords in URL: Some keywords occur more than usual in a phishing URL. The higher is the number of phishing keywords used in a URL, the more suspicious is the webpage. The keywords introduced in [20] are used in this paper. $F13$ is the number of phishing keywords that exist in the path segment of URL.

Feature 14. <script> resource identity: $F14$ is introduced in [2] and is applied in this paper. The attackers try to register an address whose URL, at a glance, is similar to the address of a real website; in this way, the novice user may not distinguish the difference [2]. In legitimate websites, the resource elements such as scripts are called from their own address. However, in the phishing webpage, the majority of the resources are called from the legitimate websites. $F14$ uses the Levenshtein distance [21] to evaluate the webpage "script" resource identity. The domains of the "src" attribute of all script tags are extracted here. Then for each domain, a normalize distance ($L_{normalize}$) introduced in [22] is calculated. $F14=\left(\sum_1^n L_{normalize}\right)\Big/n$ for n script tags.

Feature 15-17. <script>, <link> and <img> resources access protocol: Phishing pages and their resource elements do not use the https protocol. $F15$, $F16$, and $F17$ are defined as the features to show the rate of secure access to page links, JavaScript files, and images, respectively. If $N_{links}\neq 0$, $F15=N_{secureLinks}/N_{links}$; otherwise, $F15=0$. If $N_{scripts}\neq 0$, $F16=N_{secureScripts}/N_{scripts}$; otherwise, $F16=0$. If $N_{images}\neq 0$, $F17=N_{secureImages}/N_{images}$; otherwise, $F17=0$, where $N_{secureImages}$, $N_{secureScripts}$, and $N_{securLinks}$ are the number of image, script, and link tags that use the https protocol and N_{images}, $N_{scripts}$, and N_{links} are the number of all images, script, and link tags, respectively.

9. Login form detector

This module reduces unnecessary calculations and improves the PDWS response time. Phishers try to obtain the login details of users through a fake login form. In this module, if there is no login form, the webpage labels as legitimate; otherwise, the webpage moves on to the webpage feature extractor module. Here, to verify the login form, the same algorithm as [15] is used.

10. Phishing classifier

In this paper, the Extreme Learning Machine (ELM), a well-known data classification technique, is applied to classify the webpage features. ELM is a learning algorithm for single hidden layer feed forward Neural Networks (SLFNs). The least square method is used to train the networks [23]. ELM randomly assigns values for input weights and biases, and analytically determines the output weights of SLFNs [23]. The learning speed of ELM is extremely fast. In the kernel-based ELM, the kernel matrix determines the hidden layer feature mapping. In this paper, a webpage is only considered legitimate or phishing. It is naturally a binary classification problem. ELM produces output in two classes: +1 means phishing, and 0 means legitimate. RBF-kernel ELM is utilized here.

11. Experiments

The feature selection method is implemented in Matlab, and the anti-phishing system is implemented in Java. All implementations are carried out using a system with four 3GHz processors and 12 GB RAM. The system is evaluated using True Positive Rate (TPR), False Positive Rate (FPR), and Accuracy (ACC). TPR/FPR indicates the percentage of correctly/wrongly classified phishing webpages, respectively, as:

$$TPR=\frac{TP}{P},\ FPR=\frac{FP}{L} \tag{11}$$

ACC indicates the degree of closeness between measurements of classified webpages and sum of the actual phishing and legitimate webpages as:

$$ACC = \frac{TP + TN}{P + L} \quad (12)$$

where TP / FP is the number of correctly/wrongly classified phishing, TN is the number of correctly classified legitimate, and P / L is the number of phishing/legitimate pages.

11.1. Description of data

Real-world data are collected from 5000 live English phishing and legitimate pages from November 2015 to January 2016. This data consists of 2500 legitimate and 2500 phishing pages. Legitimate URLs are obtained from four sources: 1600, 330, 220, 350 webpages from Google's top 1000 most visited sites [24], Alexa's top sites [25], Moz's top 500 sites [26], and Netcraft most visited website list [27]. Phishing URLs are collected from Phishtank database [28].

11.2. Experimental results

In all experiments, 80% and 20% of the dataset are considered as train and test data, respectively. The training data includes 2000 phishing and 2000 legitimate webpages. The testing data includes 500 phishing and 500 legitimate webpages. 10-fold cross-validation is applied to tune and select all parameter and feature subsets. Since the extraction time of the features is not the same, the average of the feature extraction time is considered as the feature cost. The costs of some features are shown in table 1.

11.2.1. Evaluation of CBMRMR in PWDS without pre-filter module

In this experiment, the performance of the suggested feature selection CBMRMR based on the proposed system without pre-filter module is evaluated, and selection of relevant features is assessed. In the pre-processing phase, the CBMRMR and MRMR methods are applied on the dataset, and different numbers of features are selected. Table 2 shows the impact of CBMRMR and suggested anti-phishing system on the accuracy and average runtime.

Table 1. The cost of some features.

Feature	Extraction time (ms)	Feature	Extraction time (ms)	Feature	Extraction time (ms)
Feature 1	28.2465	Feature 2	32.9208	Feature 12	0.0048
Feature 10	0.00025	Feature 8	8.8670	Feature 4	0.00075

Table 2. Evaluation of suggested anti-phishing system for several values of γ .

			Number of selected features							
			5		10		15		20	
		γ	ACC %	Avg. runtime *Ms*	ACC %	Avg. runtime *Ms*	ACC %	Avg. runtime *ms*	ACC %	Avg. runtime *Ms*
Method	MRMR	-	94.6	5355.42	96.4	1300.3	96.4	3986.9	97.3	3996.8
	CBMRMR	100	95.9	94.896	95.3	109.67	96.6	121.76	97.3	126.1
		300	95.5	94.896	95.3	109.67	96.6	121.74	97.3	126.21
		500	95.6	535.42	96.1	550.19	96.8	552.33	97.6	552.46
		700	95.6	535.42	96.1	550.19	96.8	552.33	97.6	562.39
		900	95.6	535.42	96.1	550.19	96.8	552.33	97.6	562.39

In the next experiment, CBMRMR is compared with the method proposed in [29]. In this paper, the method is called "CanedoMRMR". CBMRMR and CanedoMRMR are applied to select the best relevant features. First, CanedoMRMR is used for several values of λ (λ is a parameter introduced to weight the influence of the cost CanedoMRMR [29]). Secondly, different numbers of feature subsets are chosen from the output of CanedoMRMR.

Thirdly, for each subset, the total cost of the features in the subset is measured and set as γ in CBMRMR. Finally, the CanedoMRMR and CBMRMR outputs are sent to the proposed PWDS, and accuracy and average runtime of the system are calculated. Table 3 shows the detection time of a webpage for several values of λ .

If λ = 0.2 and the five features are selected by CanedoMRMR, sum of the time of features is 27.006 ms. Figure 2 shows the classification accuracy for some of the selected features by CBMRMR and CanedoMRMR. According to table 3 and figure 2, by reducing the value of λ in CanedoMRMR, the influence of the feature cost is reduced and the accuracy of PWDS is increased. However, it causes an increase in the value of γ in

CBMRMR. This method selects more effective features than CanedoMRMR with a higher accuracy. The 17 features explained in Section 8 are selected by CBMRMR in this paper. The contribution of the individual features in ELM classification is demonstrated in figure 3.

The statistics show that the proposed features (features 1 and 2) outperform all the other features used in the system with an AUC of over 0.85.

Table 3. Detection time of a webpage with different λ in ms.

Number of selected feature	λ = 0.002	0.004	0.006	0.02	0.04	0.06	0.2	0.4	0.6	0.8	1	2
5	41.538	41.537	41.537	27.006	27.006	27.006	27.006	26.698	26.699	26.689	26.689	26.599
10	43.032	43.031	43.031	28.418	28.418	28.418	27.102	27.102	26.785	26.764	26.698	26.698
15	43.856	43.856	43.856	34.235	34.235	34.235	27.103	27.103	27.102	26.763	26.699	26.699
20	58.233	48.298	43.955	34.334	34.334	34.334	27.585	27.105	27.103	27.103	26.787	26.722
25	58.239	58.239	43.961	34.340	34.340	34.340	27.732	27.151	27.151	27.162	26.845	26.828
30	58.291	58.291	53.959	34.414	34.414	34.414	27.813	27.812	27.812	27.812	27.642	27.642

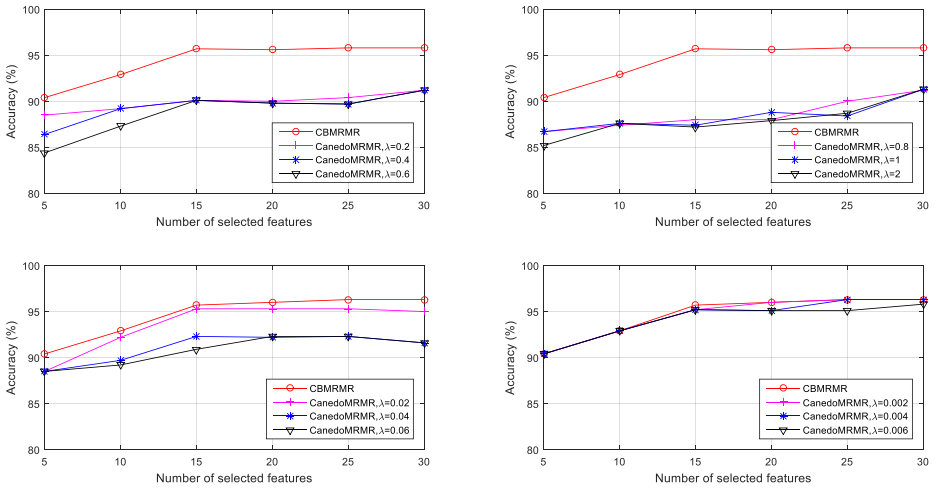

Figure 2. Comparison of CBMRMR and CanedoMRMR methods for several values of λ.

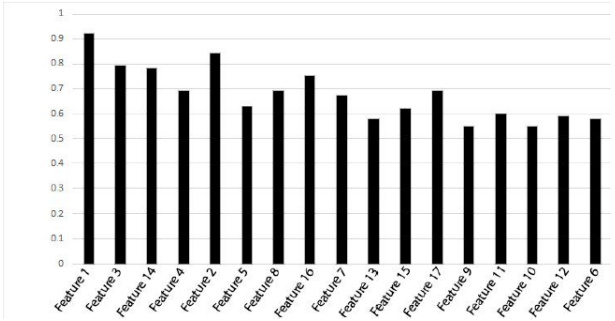

Figure 3. Area under ROC curve for individual features.

11.2.2. Evaluation of proposed system

In this section, the final evaluation of the system with and without the concerning system response time is demonstrated. In the case of evaluation without concerning the system response time, the most superior features are extracted from the final result of MRMR feature selection, while in the other case, the relevant features are selected by CBMRMR (the features mentioned in part 8). Then ACC, TPR, FPR, and average system response time are measured.

In accordance with table 4, in the second case, the system has a significant effect on the average system response time.

Table 4. Comparison of suggested test results with those of other anti-phishing methods.

Method	ACC (%)	TPR (%)	FPR (%)	Avg. Runtime (*ms*)
Phishtackle [3]	85.4	85.8	15	11523
CEADPW [13]	77.5	76.2	21.2	35337.7
MLPDHA [30]	91.46	90.9	7.6	12538.14
Our method without cost	97.3	97.6	3	3980.75
Our method with cost	97.6	97.6	2.4	120.07

12. Conclusion

In this paper, a system is proposed to detect phishing pages with a high accuracy and a short response time. Two new features are presented: one determines the relationship between the content and the address of the page, and the other calculates the relationship between the text anchor and the page address using semantic relations. In order to select the best features, a cost-based MRMR called CBMRMR is suggested. This algorithm considers the total cost of the selected features by MRMR till now. The features selected are used in the proposed PWDS, and the ELM classifier determines the page label. The results of the experimental study show that our phishing detection system has a high accuracy of 97.6% and a short system response time of 120.07 ms. In comparison with similar PWDS systems, the performance of the proposed system improves dramatically, especial in terms of system response time. The phishing problem can be solved using multi-objective optimization as a future work.

References

[1] Noghani, F. F. & Moattar, M. H. (2017). Ensemble classification and extended feature selection for credit card fraud detection. Journal of AI and Data Mining, vol. 5, no. 2, pp. 235-243.

[2] Moghimi, M. & Yazdani, A. V. (2016). New rule-based phishing detection method. Expert Systems with Applications, vol. 53, pp. 231-242.

[3] Ghowtham, R. & Krishnamurthi, I. (2014). PhishTackle- a web services architecture for anti-phishing. Cluster Computing, vol. 17, no. 3, pp. 1051-1068.

[4] Arachchilage, N. A., Love, S. & Beznosov, K. (2016). Phishing threat avoidance behaviour: An empirical investigation. Computers in Human Behavior, vol. 60, pp.185-197.

[5] Tan, C. L., Chiew, K. L. & Wong, K. (2016). PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. Decision Support Systems, vol. 88, pp. 18-27.

[6] Bose, J. & Leung, A. C. M. (2014). Do phishing alerts impact global corporations? A firm value analysis. Decision Support System, vol. 64, pp. 67-78.

[7] Kang, J. & Lee, D. (2007). Advanced white list approach for preventing access to phishing sites. International Conference on Convergence Information Technology, Gyeongju, Korea, 2007.

[8] Prakash, P., Kumar, M., Kompella, R.& Gupta, M. (2010). PhishNet: predictive blacklisting to detect phishing attacks. IEEE INFOCOM, San Diego, USA, 2010.

[9] Fu, A. Y., Wenyin, L. & Deng, X. (2006). Detecting phishing web pages with visual similarity assessment based on earth mover's distance. IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 4, pp. 301-311.

[10] Weynin, L., Huang, G., Xiaoyue, L., Min, Z. & Deng, X. (2005). Detection of phishing webpages based on visual similarity. Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, Chiba, Japan, 2005.

[11] Medvet, E., Kirda, E. & Kruegel, C. (2008). Visual-similarity-based phishing detection. 4th International Conference on Security and Privacy in Communication, Istanbul, Turkey, 2008.

[12] Zhang, Y., Hong, L. & Cranor, L. F. (2007). Cantina: a content-based approach to detecting phishing web sites. 16th International Conference on World Wide Web, Banff, Canada, 2007.

[13] Ghowtham, R. & Krishnamurthi, I. (2014). A comprehensive and efficacious architecture for detecting phishing webpages. Computers & Security, vol. 40, pp. 23-37.

[14] He, M., Horng, S. J., Fan, P., Khan, M. K., Run, R. S., Lai, J. L., Chen, R. J. & Sutanto, A. (2011). An efficient phishing webpage detector. Expert Systems with Application, vol. 38, no. 10, pp. 12018-12027.

[15] Xiang, G., Hong, J., Rose, C. & Cranor, L. (2011). Cantina+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security, vol. 14, no. 2.

[16] Abdelhamid, N., Ayesh, A. & Thabtah, F. (2014). Phishing detection based Associative

Classification data mining. Expert Systems with Applications, vol. 41, no. 13, pp. 5948-5959.

[17] Ding, C. & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology, vol. 3, no. 2, pp. 185-205.

[18] Wu, Z. & Palmer, M. (1994). Verbs semantics and lexical selection. 32nd Annual Meeting on Association for Computational Linguistics, Las Cruces, New Mexico, 1994.

[19] Vrîncianu, M. & Popa, L. (2010). Considerations regarding the security and protection of e-banking services consumers' interests. The Amfiteatru Economic Journal, vol. 12, no. 28, pp. 388-403.

[20] Garera S, Provos N, Chew M, Rubin AD. A framework for detection and measurement of phishing attacks. In: The 2007 ACM workshop on Recurring Malcode; 29 October-02 November 2007; Alexandria, VA, USA: ACM. pp.1-8.

[21] Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, vol. 10, no. 8, pp. 707-710.

[22] Fu, A. Y. (2006). Web identity security: advanced phishing attacks and counter measures. PhD, City University of Hong Kong, Kowloon Tong, Kowloon, Hong Kong, 2006.

[23] Huang, G. B., Zhu, Q. Y. & Siew, C. K. (2006). Extreme learning machine: Theory and applications. Neurocomputing, vol. 70, pp. 481-501.

[24] Google's top 1000 most visited sites (2015), Available: http://www.google.com/adplanner/static/top1000/.

[25] Alexa's top sites (2015), Available: http://www.alexa.com/topsites/.

[26] Moz's top 500 sites (2015), Available: https://moz.com/top500/.

[27] Netcraft most visited websites list (2015), Available: http://toolbar.netcraft.com/stats/topsites/.

[28] Phishtank database (2015), Available: http://www.phishtank.com/.

[29] Bolón-Canedo, V., Porto-Díaz, I., Sánchez-Maroño, N. & Alonso-Betanzos, A. (2014). A framework for cost-based feature selection. Pattern Recognition, vol. 47, no. 7, pp. 2481-2489.

[30] Jain, A. K. & Gupta, B. B. (2018). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing, pp. 1-14. https://doi.org/10.1007/s12652-018-0798-z.

نشریه هوش مصنوعی و داده کاوی

ارائه معماری جدید برای شناسایی صفحات فیشینگ با استفاده از انتخاب ویژگی مبتنی بر هزینه

عاطفه زنگویی[1]، ولی درهمی[1] و فاطمه جمشیدی[2،*]

[1] گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه یزد، یزد، ایران.

[2] گروه مهندسی برق، دانشکده مهندسی، دانشگاه فسا، فسا، ایران.

ارسال ۲۰۱۸/۰۶/۲۳؛ بازنگری ۲۰۱۹/۰۱/۲۱؛ پذیرش ۲۰۱۹/۰۳/۰۷

چکیده:

یکی از تکنیک‌های فریب‌دهنده برای بدست آوردن اطلاعات شخصی کاربران، فیشینگ است. یک سیستم تشخیص صفحات فیشینگ، صفحات فیشینگ را بر اساس ویژگی‌های استخراج شده از صفحات، شناسایی می‌کند. بنابراین انتخاب ویژگی‌های موثر کارایی این سیستم را بهبود می‌بخشد. کارایی یک سیستم به دقت تشخیص و زمان پاسخ سیستم وابسته است. زمان اصلی مصرف شده توسط این سیستم مربوط به زمان ویژگی‌هایی است که از صفحات استخراج و به عنوان هزینه ویژگی‌ها تلقی می‌گردد. در این مقاله دو ویژگی جدید ارائه می‌شود که ارتباط میان محتوا و آدرس صفحات را با استفاده از شباهت معنایی محاسبه می‌کنند. ویژگی‌های ارائه شده از نتیجه موتورهای جستجو و پایگاه داده Whois که زمان‌بر هستند، استفاده نمی‌کنند. بنابراین زمان استخراج ویژگی‌ها به طور چشم‌گیری کاهش می‌یابد. به منظور کاهش نرخ مثبت کاذب و محاسبات غیر ضروری، از پیش فیلتر شناسایی فرم ورود درون صفحات استفاده می‌شود. در این مقاله به منظور انتخاب موثرترین ویژگی‌ها، یک الگوریتم انتخاب ویژگی مبتنی بر هزینه ارائه می‌شود. ویژگی‌های انتخاب شده در سیستم پیشنهاد شده به‌کار می‌روند. به منظور دسته‌بندی صفحات، از الگوریتم ماشین یادگیری افراطی استفاده می‌شود. نتایج تجربی نشان می‌دهد که سیستم پیشنهادی دقت بالای ۹۷/۶ درصد و میانگین زمان اجرای پایین ۱۲۰/۰۷ میلی ثانیه دارد.

کلمات کلیدی: انتخاب ویژگی مبتنی بر هزینه، ماشین یادگیری افراطی، فیشینگ، شباهت معنایی، فراوانی کلمه و معکوس فراوانی متن.